```
Imports System.Drawing.Drawing2D

Public Class RunChart
    Inherits System.Windows.Forms.UserControl

#Region "Variables and Enumerations"
    Private privatetotalSamples As Integer = 50
    Private privateSamples(50) As Single
    Private privateCurrentSampleNumber As Integer = 0

    Private privatePanelHeight As Integer = CInt(Me.Height - 4)
    Private privatePanelWidth As Integer = CInt(Me.Width - 4)
    Private privateLineColor As System.Drawing.Color = Color.Gold
    Private privatePreviousLineColor As System.Drawing.Color = Color.Gold
    Private privatePanelColor As System.Drawing.Color = Color.Black
    Private privatePanelBackColor As System.Drawing.Color = Color.Black
    Private privateUSL As Single = 0.04
    Private privateUSL_Color As Color = Color.Red
    Private privateUCL As Single = 0.025
    Private privateUCL_Color As Color = Color.Gold
    Private privateNominal As Single = 0.0
    Private privateNominal_Color As Color = Color.Lime
    Private privateLCL As Single = -0.025
    Private privateLCL_Color As Color = Color.Gold
    Private privateLSL As Single = -0.04
    Private privateLSL_Color As Color = Color.Red
    Private privateRunChartValue As Single = 0.0
    Private privateRunChartPreviousValue As Integer = 0.0
    Private privateSpaceLeft As Integer = 5
    Private privateXAxisDivision As Integer = CInt((privatePanelWidth - privateSpaceLeft) /
privatetotalSamples)
    Private privateX1 As Integer
    Private privateY1 As Integer
    Private privateX2 As Integer
    Private privateY2 As Integer
    Private privateOldX1 As Integer
    Private privateOldY1 As Integer
    Private privateOldX2 As Integer
    Private privateOldY2 As Integer
    Private privateRunChartRange As Double = 0.1
    Private isClickable As Boolean = True

    Private privateControlCause As Single = 0.0

#End Region

#Region "Control Properties"

    Public Property RunChartSamplesValue() As Single
        Get
            Return privateSamples(privateCurrentSampleNumber)
        End Get
```

```
    Set(ByVal value As Single)
        'If (value <= privateUSL) And (value >= privateLSL) Then
        If (privateCurrentSampleNumber < privatetotalSamples) Then
            privateSamples(privateCurrentSampleNumber) = value
            privateCurrentSampleNumber = privateCurrentSampleNumber + 1
            Draw_RunChart() 'draw only run chart

        ElseIf (privateCurrentSampleNumber = privatetotalSamples) Then
            Dim f_i As Integer
            For f_i = 0 To privatetotalSamples - 2
                privateSamples(f_i) = privateSamples(f_i + 1)
            Next f_i
            privateSamples(privatetotalSamples - 1) = value

            Me.Invalidate() 'Draw complete run chart including tolerence marker lines
        End If

    End Set
End Property

Public Property RunChartCurrentSampleCount() As Integer
    Get
        Return privateCurrentSampleNumber
    End Get

    Set(ByVal value As Integer)
        privateCurrentSampleNumber = value
        Me.Invalidate()
    End Set
End Property

Public Property RunChartTotalSampleCount() As Integer
    Get
        Return privatetotalSamples
    End Get
    Set(ByVal value As Integer)
        If (value <= 100) Then
            privatetotalSamples = value
        End If
    End Set
End Property


Public Property RunChartUSL() As Single
    Get
        Return privateUSL
    End Get
    Set(ByVal value As Single)
        'If (value >= privateUCL) Then
        privateUSL = value
        Calculate_Range()
```

```
        'End If
    End Set
End Property

Public Property RunChartUSLColor() As Color
    Get
        Return privateUSL_Color
    End Get
    Set(ByVal value As Color)
        privateUSL_Color = value
        Me.Invalidate()
    End Set
End Property

Public Property RunChartUCL() As Single
    Get
        Return privateUCL
    End Get
    Set(ByVal value As Single)
        If (value <= privateUSL) Then
            privateUCL = value
            Calculate_Range()

        End If
    End Set
End Property

Public Property RunChartUCLColor() As Color
    Get
        Return privateUCL_Color
    End Get
    Set(ByVal value As Color)
        privateUCL_Color = value
        Me.Invalidate()
    End Set
End Property

Public Property RunChartNominal() As Single
    Get
        Return privateNominal
    End Get
    Set(ByVal value As Single)
        If (value <= privateUSL) And (value <= privateUCL) Then
            privateNominal = value
            Calculate_Range()

        End If
    End Set
End Property

Public Property RunChartNominalColor() As Color
    Get
```

```
        Return privateNominal_Color
    End Get
    Set(ByVal value As Color)
        privateNominal_Color = value
        Me.Invalidate()
    End Set
End Property

Public Property RunChartLCL() As Single
    Get
        Return privateLCL
    End Get
    Set(ByVal value As Single)
        If (value <= privateNominal) Then
            privateLCL = value
            Calculate_Range()

        End If
    End Set
End Property

Public Property RunChartLCLColor() As Color
    Get
        Return privateLCL_Color
    End Get
    Set(ByVal value As Color)
        privateLCL_Color = value
        Me.Invalidate()
    End Set
End Property

Public Property RunChartLSL() As Single
    Get
        Return privateLSL
    End Get
    Set(ByVal value As Single)
        If (value <= privateLCL) Then
            privateLSL = value
            Calculate_Range()

        End If
    End Set
End Property

Public Property RunChartLSLColor() As Color
    Get
        Return privateLSL_Color
    End Get
    Set(ByVal value As Color)
        privateLSL_Color = value
        Me.Invalidate()
    End Set
```

```
    End Property

    Public Property Clickable() As Boolean
        Get
            Return isClickable
        End Get
        Set(ByVal value As Boolean)
            isClickable = value
            Me.Invalidate()
        End Set
    End Property

    Public Property ControlCause() As Single
        Get
            Return privateControlCause
        End Get
        Set(ByVal value As Single)
            privateControlCause = value
            Me.Invalidate()
        End Set
    End Property


#End Region

#Region "Drawing Functions"

    Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)

        Dim rect As System.Drawing.Rectangle = e.ClipRectangle
        Dim g As Graphics = e.Graphics
        Dim mainPen As New Pen(Color.Black)

        Draw_Horizontal_Axis()
        Draw_RunChart()
    End Sub

#End Region

#Region "Event Handlers"
    Protected Overrides Sub OnMouseDown(ByVal e As System.Windows.Forms.MouseEventArgs)
        If isClickable Then
            If e.Button = Windows.Forms.MouseButtons.Left Then

            Else
                'Score -= 1
            End If
        End If
    End Sub
#End Region

    Protected Overrides Sub Finalize()
```

```
        MyBase.Finalize()
    End Sub

    Private Sub Calculate_Range()

        privateRunChartRange = (privateUSL - privateLSL) * 2

        Me.Invalidate()

    End Sub

    Private Sub Draw_Horizontal_Axis()
        Dim myPen As New System.Drawing.Pen(System.Drawing.Color.Crimson)
        Dim formGraphics As System.Drawing.Graphics
        formGraphics = Me.CreateGraphics()

        Dim foo_float As Single
        Dim foo_pos1 As Integer


        privatePanelHeight = CInt(Me.Height - 4)
        privatePanelWidth = CInt(Me.Width - 4)

        myPen.Width = 2

        myPen.Color = privateUSL_Color
        foo_float = ((privateUSL - privateNominal) / (privateRunChartRange / 2.0)) * ((Me.Height) /
2.0)
        foo_pos1 = CInt(foo_float)
        formGraphics.DrawLine(myPen, CInt(privateSpaceLeft), CInt((privatePanelHeight / 2) -
foo_pos1), CInt((Me.Width) - privateSpaceLeft), CInt((privatePanelHeight / 2) - foo_pos1)) 'Draw
USL Line

        myPen.Color = privateUCL_Color
        foo_float = ((privateUCL - privateNominal) / (privateRunChartRange / 2.0)) * ((Me.Height) /
2.0)
        foo_pos1 = CInt(foo_float)
        formGraphics.DrawLine(myPen, CInt(privateSpaceLeft), CInt((privatePanelHeight / 2) -
foo_pos1), CInt((Me.Width) - privateSpaceLeft), CInt((privatePanelHeight / 2) - foo_pos1)) 'Draw
UCL Line

        myPen.Color = privateNominal_Color
        formGraphics.DrawLine(myPen, CInt(privateSpaceLeft), CInt((privatePanelHeight / 2)),
CInt((Me.Width) - privateSpaceLeft), CInt((privatePanelHeight / 2))) 'Draw Nominal Value Center
Line

        Dim f_i As Integer

        privateXAxisDivision = (Me.Width - privateSpaceLeft) / (privatetotalSamples + 1)
        For f_i = 0 To privatetotalSamples
            formGraphics.DrawLine(myPen, CInt((privateSpaceLeft + (privateXAxisDivision * f_i))),
CInt((privatePanelHeight / 2) + 3), CInt((privateSpaceLeft + (privateXAxisDivision * f_i))),
```

```
CInt((privatePanelHeight / 2) - 3)) 'Draw Nominal Value Center Line
        Next f_i

        myPen.Color = privateLCL_Color
        foo_float = (((privateLCL - privateNominal) / ((privateRunChartRange) / 2.0)) *
((Me.Height) / 2.0))
        foo_pos1 = CInt(foo_float)
        formGraphics.DrawLine(myPen, CInt(privateSpaceLeft), CInt((privatePanelHeight / 2) -
foo_pos1), CInt((Me.Width) - privateSpaceLeft), CInt((privatePanelHeight / 2) - foo_pos1)) 'Draw
LCL Line

        myPen.Color = privateLSL_Color
        foo_float = (((privateLSL - privateNominal) / ((privateRunChartRange) / 2.0)) *
((Me.Height) / 2.0))
        foo_pos1 = CInt(foo_float)
        formGraphics.DrawLine(myPen, CInt(privateSpaceLeft), CInt((privatePanelHeight / 2) -
foo_pos1), CInt((Me.Width) - privateSpaceLeft), CInt((privatePanelHeight / 2) - foo_pos1)) 'Draw
LSL Line

        myPen.Dispose()
        formGraphics.Dispose()

    End Sub

    Private Function GetColorOFLine(ByRef fii As Integer) As Color
        If privateSamples(fii) >= privateUSL Then
            GetColorOFLine = privateUSL_Color
        ElseIf privateSamples(fii) <= privateLSL Then
            GetColorOFLine = privateLSL_Color
        ElseIf privateSamples(fii) >= privateUCL AndAlso privateSamples(fii) <= privateUSL Then
            GetColorOFLine = Color.WhiteSmoke 'privateUCL_Color
        ElseIf privateSamples(fii) <= privateLCL AndAlso privateSamples(fii) >= privateLSL Then
            GetColorOFLine = Color.WhiteSmoke 'privateLCL_Color
        ElseIf privateSamples(fii) <= privateUCL AndAlso privateSamples(fii) >= privateLCL Then
            GetColorOFLine = Color.WhiteSmoke 'privateNominal_Color
        Else
            GetColorOFLine = Color.WhiteSmoke
        End If
    End Function

    Private Sub Draw_RunChart()
        Dim myPen As New System.Drawing.Pen(System.Drawing.Color.WhiteSmoke)
        Dim formGraphics As System.Drawing.Graphics
        formGraphics = Me.CreateGraphics()
        Dim X1, Y1, X2, Y2 As Integer
        Dim Current_Reading As Single
        Dim Current_Position As Integer

        If (privateCurrentSampleNumber = 0) Then
            X1 = privateSpaceLeft + (privateXAxisDivision * privateCurrentSampleNumber)
```

```
        Current_Reading = ((privateSamples(privateCurrentSampleNumber) - privateNominal) /
(privateRunChartRange / 2.0)) * ((Me.Height) / 2.0)
        Current_Position = CInt(Current_Reading)

        Y1 = (privatePanelHeight / 2) - Current_Position

        privateOldX1 = X1
        privateOldY1 = Y1

      ElseIf (privateCurrentSampleNumber > 0) Then

        X1 = privateSpaceLeft

        Current_Reading = ((privateSamples(0) - privateNominal) / (privateRunChartRange / 2.0))
* ((Me.Height) / 2.0)
        Current_Position = CInt(Current_Reading)
        Y1 = (privatePanelHeight / 2) - Current_Position

        myPen.Width = 2

        If (Y1 <= (Me.Height)) Then 'Draw the point if the point lies in the graphics area other wise
draw only line
           formGraphics.DrawEllipse(myPen, X1, Y1, 3, 3)
           privateOldY1 = Y1
        Else
           formGraphics.DrawEllipse(myPen, X1, CInt((Me.Height)), 3, 3)
           privateOldY1 = CInt((Me.Height))
        End If


        privateOldX1 = X1


        Dim f_i As Integer
        For f_i = 0 To privateCurrentSampleNumber - 1
           X2 = privateSpaceLeft + (privateXAxisDivision * f_i)

           Current_Reading = ((privateSamples(f_i) - privateNominal) / (privateRunChartRange /
2.0)) * ((Me.Height) / 2.0)
           Current_Position = CInt(Current_Reading)

           Y2 = (privatePanelHeight / 2) - Current_Position



           myPen.Color = GetColorOFLine(f_i)
           myPen.Width = 2

           formGraphics.DrawLine(myPen, privateOldX1, privateOldY1, X2, Y2) 'Draw Line
           'If (Y2 <= (Me.Height) / 2.0) Then 'Draw the point if the point lies in the graphics area
other wise draw only line
           myPen.Color = Color.WhiteSmoke
```

```
            formGraphics.DrawEllipse(myPen, X2, Y2, 3, 3)
            'Else
            '    formGraphics.DrawEllipse(myPen, X2, CInt((Me.Height)), 3, 3)
            'End If

            privateOldX1 = X2
            privateOldY1 = Y2

        Next f_i

    End If

    myPen.Dispose()
    formGraphics.Dispose()

  End Sub

End Class
```